METHOD AND SYSTEM FOR CAPTURING AND BYPASSING MEMORY TRANSACTIONS IN A HUB-BASED MEMORY SYSTEM

TECHNICAL FIELD

This invention relates to computer systems, and, more particularly, to a computer system including a system memory having a memory hub architecture.

BACKGROUND OF THE INVENTION

Computer systems use memory devices, such as dynamic random access memory ("DRAM") devices, to store data that are accessed by a processor. These memory devices are normally used as system memory in a computer system. In a typical computer system, the processor communicates with the system memory through a processor bus and a memory controller. The processor issues a memory request, which includes a memory command, such as a read command, and an address designating the location from which data or instructions are to be read. The memory controller uses the command and address to generate appropriate command signals as well as row and column addresses, which are applied to the system memory. In response to the commands and addresses, data are transferred between the system memory and the processor. The memory controller is often part of a system controller, which also includes bus bridge circuitry for coupling the processor bus to an expansion bus, such as a PCI bus.

Although the operating speed of memory devices has continuously increased, this increase in operating speed has not kept pace with increases in the operating speed of processors. Even slower has been the increase in operating speed of memory controllers coupling processors to memory devices. The relatively slow speed of memory controllers and memory devices limits the data bandwidth between the processor and the memory devices.

In addition to the limited bandwidth between processors and memory devices, the performance of computer systems is also limited by latency problems that increase the time required to read data from system memory devices. More specifically,

when a memory device read command is coupled to a system memory device, such as a synchronous DRAM ("SDRAM") device, the read data are output from the SDRAM device only after a delay of several clock periods. Therefore, although SDRAM devices can synchronously output burst data at a high data rate, the delay in initially providing the data can significantly slow the operating speed of a computer system using such SDRAM devices.

One approach to alleviating the memory latency problem is to use multiple memory devices coupled to the processor through a memory hub. In a memory hub architecture, a system controller or memory controller is coupled over a high speed data link to several memory modules. Typically, the memory modules are coupled in a point-to-point or daisy chain architecture such that the memory modules are connected one to another in series. Thus, the memory controller is coupled to a first memory module over a first high speed data link, with the first memory module connected to a second memory module through a second high speed data link, and the second memory module coupled to a third memory module through a third high speed data link, and so on in a daisy chain fashion.

Each memory module includes a memory hub that is coupled to the corresponding high speed data links and a number of memory devices on the module, with the memory hubs efficiently routing memory requests and responses between the controller and the memory devices over the high speed data links. Computer systems employing this architecture can have a higher bandwidth because a processor can access one memory device while another memory device is responding to a prior memory access. For example, the processor can output write data to one of the memory devices in the system while another memory device in the system is preparing to provide read data to the processor. Moreover, this architecture also provides for easy expansion of the system memory without concern for degradation in signal quality as more memory modules are added, such as occurs in conventional multi drop bus architectures.

Although computer systems using memory hubs may provide superior performance, they nevertheless may often fail to operate at optimum speeds for a variety of reasons. For example, even though memory hubs can provide computer systems with

a greater memory bandwidth, they still suffer from latency problems of the type described above. More specifically, although the processor may communicate with one memory device while another memory device is preparing to transfer data, it is sometimes necessary to receive data from one memory device before the data from another memory device can be used. In the event data must be received from one memory device before data received from another memory device can be used, the latency problem continues to slow the operating speed of such computer systems.

Another factor that can reduce the speed of memory transfers in a memory hub system is the delay in forwarding memory requests from one memory hub to another. For example, in a system including five memory modules (*i.e.* five memory hubs with one per module), a memory request to read data from the fifth module that is farthest "downstream" from the memory controller will be delayed in being applied to the fifth memory module due to the intervening delays introduced by the first through fourth memory modules in processing and forwarding the memory request. Moreover, where the applied command is a command to read data from a memory module, the longer the delay in applying the read command to the memory module the longer it will take for the memory module to provide the corresponding read data, increasing the latency of the module. The farther downstream a memory module the longer the delay in applying a memory request and the greater the latency in reading data, lowering the bandwidth of the system memory.

Still another concern with a memory hub architecture is the complexity of the circuitry required to form each memory hub. Complex circuitry increases the cost of each memory hub, which increases the cost of each memory module and the overall cost of system memory as modules are added. As the functions each memory hub must perform increase, the complexity of the circuitry increases accordingly. In one implementation of a memory hub architecture, each hub must determine whether a given memory request is directed to that module. If the memory request is directed to the module, the hub processes the request, and if not the request is forwarded to the next downstream hub. A variety of other functions must also be performed by each memory

hub, such as generating all the control, data, and address signals for accessing the memory devices on the memory module.

There is therefore a need for a computer architecture that provides the advantages of a memory hub architecture and also minimizes delays in processing downstream memory requests to provide a high bandwidth system memory.

SUMMARY OF THE INVENTION

According to one aspect of the present invention, a memory hub, includes a reception interface that receives data words and captures the data words in response to a first clock signal in a first time domain. The interface also provides groups of the captured data words on an output in response to a second clock signal in a second time domain. A transmission interface is coupled to the reception interface to receive the captured data words and captures the data words in response to a third clock signal in the first time domain. This interface provides the captured data words on an output. Local control circuitry is coupled to the output of the reception interface to receive the groups of data words and develops memory requests corresponding to the groups of data words.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a computer system including a system memory having a high-bandwidth memory hub architecture according to one example of the present invention.

Figure 2 is a block diagram illustrating the memory hubs contained in the memory modules in the system memory of Figure 1 according to one example of the present invention.

Figure 3 is a more detailed block diagram the memory hubs of Figure 2 according to one example of the present invention.

Figure 4 is signal timing diagram illustrating the operation of the memory hub of Figure 3 in capturing and forwarding downstream memory requests.

DETAILED DESCRIPTION OF THE INVENTION

A computer system 100 according to one example of the present invention is shown in Figure 1. The computer system 100 includes a system memory 102 having a memory hub architecture that efficiently forwards and processes downstream memory requests to provide a high bandwidth memory system, as will be explained in more detail below. The computer system 100 includes a processor 104 for performing various computing functions, such as executing specific software to perform specific calculations or tasks. The processor 104 includes a processor bus 106 that normally includes an address bus, a control bus, and a data bus. The processor bus 106 is typically coupled to cache memory 108, which, as previously mentioned, is usually static random access memory ("SRAM"). Finally, the processor bus 106 is coupled to a system controller 110, which is also sometimes referred to as a "North Bridge" or "memory controller."

The system controller 110 serves as a communications path to the processor 104 for a variety of other components. More specifically, the system controller 110 includes a graphics port that is typically coupled to a graphics controller 112, which is, in turn, coupled to a video terminal 114. The system controller 110 is also coupled to one or more input devices 118, such as a keyboard or a mouse, to allow an operator to interface with the computer system 100. Typically, the computer system 100 also includes one or more output devices 120, such as a printer, coupled to the processor 104 through the system controller 110. One or more data storage devices 124 are also typically coupled to the processor 104 through the system controller 110 to allow the processor 104 to store data or retrieve data from internal or external storage media (not shown). Examples of typical storage devices 124 include hard and floppy disks, tape cassettes, and compact disk read-only memories (CD-ROMs).

The system controller 110 is further coupled to the system memory 102, which includes several memory modules 130a,b...n. The memory modules 130 are coupled in a point-to-point or daisy chain architecture through respective high speed links 134 coupled between the modules and the system controller 110. The high-speed links 134 may be optical, RF, or electrical communications paths, or may be some other

suitable type of communications paths, as will be appreciated by those skilled in the art. In the event the high-speed links 134 are implemented as optical communications paths, each optical communication path may be in the form of one or more optical fibers, for example. In such a system, the system controller 110 and the memory modules 130 will each include an optical input/output port or separate input and output ports coupled to the corresponding optical communications paths.

Although the memory modules 130 are shown coupled to the system controller 110 in a daisy architecture, other topologies may also be used, such as a switching topology in which the system controller 110 is selectively coupled to each of the memory modules 130 through a switch (not shown), or a multi-drop architecture in which all of the memory modules 130 are coupled to a single high-speed link 134. Other topologies that may be used, such as a ring topology, will be apparent to those skilled in the art.

Each of the memory modules 130 includes a memory hub 140 for communicating over the corresponding high-speed links 134 and for controlling access to six memory devices 148, which are synchronous dynamic random access memory ("SDRAM") devices in the example Figure 1. However, a fewer or greater number of memory devices 148 may be used, and memory devices other than SDRAM devices may, of course, also be used. The memory hub 140 is coupled to each of the system memory devices 148 through a bus system 150, which normally includes a control bus, an address bus, and a data bus.

One example of the memory hubs 140 of Figure 1 is shown in Figure 2, which is a block diagram illustrating in more detail the memory hubs in the memory modules 130a and 130b and link interface components in the system controller 110. In the memory module 130a, the memory hub 140 includes a link interface 200 that is connected to the high-speed link 134 coupled to the system controller 110. The link interface 200 includes a downstream physical reception port 202 that receives downstream memory requests from the system controller 110 over a downstream high-speed link 204, and includes an upstream physical transmission port 206 that provides upstream memory responses to the system controller over an upstream high-speed link

208. The downstream and upstream high-speed links 204, 208 collectively form the corresponding high-speed link 134.

The system controller 110 includes a downstream physical transmission port 210 coupled to the downstream high-speed link 204 to provide memory requests to the memory module 130a, and also includes an upstream physical reception port 212 coupled to the upstream high-speed link 208 to receive memory responses from the memory module 130a. The ports 202, 206, 210, 212 and other ports to be discussed below are designated "physical" interfaces or ports since these ports are in what is commonly termed the "physical layer" of a communications system. In this case, the physical layer corresponds to components providing the actual physical connection and communications between the system controller 110 and system memory 102 (Figure 1), as will be understood by those skilled in the art.

The nature of the physical reception ports 202, 212 and physical transmission ports 206, 210 will depend upon the characteristics of the high-speed links 204, 208. For example, in the event the high-speed links 204, 208 are implemented using optical communications paths, the reception ports 202, 212 will convert optical signals received through the optical communications path into electrical signals and the transmission ports will convert electrical signals into optical signals that are then transmitted over the corresponding optical communications path.

The physical reception port 202 performs two functions on the received memory requests from the system controller 110. First, the reception port 202 captures the downstream memory request, which may be in the form of a packet and which may be referred to hereinafter as a memory request packet. The physical reception port 202 provides the captured memory request packet to local hub circuitry 214, which includes control logic for processing the request packet and accessing the memory devices 148 over the bus system 150 to provide the corresponding data when the request packet is directed to the memory module 130a.

The second function performed by the physical reception port 202 is providing the captured downstream memory request over a bypass path 216 to a downstream physical transmission port 218. The physical transmission port 218, in

turn, provides the memory request packet over the corresponding downstream high-speed link 204 to the downstream physical reception port 202 in the adjacent downstream memory module 130b. The port 202 in module 130b operates in the same way as the corresponding port in the module 130a, namely to capture the memory request packet, provide the packet to local hub circuitry 214, and provide the packet over a bypass path 216 to a downstream physical transmission port 218. The port 218 in the module 130b then operates in the same way as the corresponding port in module 130a to provide the memory request packet over the corresponding downstream high-speed link 204 to the next downstream memory module 130c (not shown in Figure 2).

The memory hub 140 in the module 130a further includes an upstream physical reception port 220 that receives memory response packets over the corresponding upstream high-speed link 208 from the upstream physical transmission port 206 in the module 130b. The reception port 220 captures the received memory request packets and provides them to the local hub circuitry 214 for processing. The precise manner in which each memory hub 140 processes the upstream response packets may vary and will not be discussed in more detail herein since it is not necessary for an understanding of the present invention.

In the system memory 102, each memory hub 140 captures the downstream memory request packets, supplies the captured packet to the local hub circuitry 214 for processing, and provides the packet to the memory hub on the next downstream memory module 130. With this approach, each memory hub 140 captures every downstream memory request packet and forwards the packet to the next downstream memory module 130. Thus, whether the packet is directed to a particular memory hub 140 or not, the packet is captured and each memory hub then processes the captured packet to determine if it is directed to that memory module 130. This approach simplifies the logic necessary to implement the local hub circuitry 214 and thus lowers the cost of each memory hub 140. This is true because the local hub circuitry 214 need not determine whether each memory request packet should be bypassed but instead all request packets are automatically bypassed. The term

"bypassed" means to provide a memory request to the next downstream memory hub 140.

The present approach also reduces the delays in forwarding memory requests to downstream memory modules 130 and thus increases the bandwidth of the system memory 102. Capturing and forwarding of the memory request packets is done by the memory hubs 140 in the physical layer and thus in the clock domain of the downstream high-speed links 204. The clock rate of the high-speed links 204 is typically very fast, and thus there is a only a very small delay introduced by each memory hub 140 in bypassing each memory request packet. In contrast, the clock rate at which the local hub circuitry 214 in each memory hub 140 operates is much slower than the clock rate of the high-speed links 204. Thus, if each memory hub 140 determined whether a given request packet should be bypassed, the overall delay introduced by that hub would be much greater and the bandwidth of the system memory 102 lowered accordingly. This may also be viewed in terms of latency of the system memory 102, with greater delays introduced by the memory hubs 140 increasing the latency of the system memory. The clock domains of the high-speed links 204 and local hub circuitry 214 will be discussed in more detail below.

The physical reception port 202, bypass path 216, and physical transmission port 218 contained in the memory hubs 140 of Figure 2 will now be discussed in more detail with reference to Figure 3, which is a more detailed functional block diagram of these components according to one example of the present invention. In the following description, these components are assumed to be contained on the memory module 130a of Figure 2. Figure 3 does not depict interface circuitry that may be contained in the memory hubs 140 as well, such as when the high-speed links 204 are optical links and the memory hubs include interface circuitry for converting optical signals into electrical signals and vice versa, as will be appreciated by those skilled in the art.

The physical reception port 202 includes a pair of input capture registers 300, 302 coupled to the downstream high-speed link 204 and clocked by a pair of complementary master reception clock signals MRCLK, MRCLK* generated locally in

the physical reception port 202. In this example, each memory request packet applied on the high-speed links 204 is formed by one or more data words DW that collectively form the packet, with the data words being applied on the high-speed data link. The MRCLK, MRCLK* signals are adjusted to have a particular phase shift relative to the data words DW, such as edges of these signals occurring in the center of a data eye of each data word, as will be understood by those skilled in the art. The capture register 300 latches a data word DW on the high-speed link 204 responsive to each rising edge of the MRCLK signal, and the capture register 302 latches a data word responsive to each rising edge of the MRCLK* signal. Each data word DW may contain data, address, or control information associated with a particular memory request.

The registers 300 and 302 apply the latched data words DW to capture first-in first-out (FIFO) buffers 304 and 306, respectively, which store the applied data words responsive to the MRCLK, MRCLK* signals. The FIFO buffers 304, 306 function to store a number of data words DW at a rate determined by the MRCLK, MRCLK* signals and thus in the clock domain of the high-speed link 204. The depth of the FIFO buffers 304, 306, which corresponds to the number of data words DW stored in the buffers, must be sufficient to provide a clock domain crossing from the high-speed clock domain of the downstream high-speed link 204 to the slower clock domain of the memory hub 140, as will be appreciated by those skilled in the art and as will be discussed in more detail below. A capture read pointer circuit 308 develops selection signals SEL responsive to a core clock signal CCLK, and applies the selection signals to control two multiplexers 310, 312. More specifically, the capture read pointer circuit 308 develops the SEL signals to selectively output groups of the data words DW stored in the FIFO buffers 304, 306 on a first-in first-out basis, where the two groups of data words from the multiplexers 310, 312 collectively correspond to a memory request packet on the high-speed link 204.

The CCLK clock is an internal clock signal of the memory hub 140 and thus defines a clock domain of the memory hub, as will be discussed in more detail below. The memory request packet from the multiplexers 310, 312 is applied to a memory controller 314 contained in the local hub circuitry 214, with the memory

controller processing the memory request packet and taking the appropriate action in response thereto. For example, the memory controller 314 controls the transfer data over the bus system 150 to and from the memory devices 148 (not shown in Figure 3) when the memory request packet is directed to the memory module 140.

The frequency of the CCLK signal is lower than the frequency of the MRCLK, MRCLK* signals, which is why the data words DW are stored in the FIFO buffers 304, 306 and then read out in groups under control of the capture read pointer circuit 308 and multiplexers 310, 312. The data words DW are thus latched by the capture registers 300, 302 and buffered in the FIFO buffers 304, 306 at a faster rate determined by the MRCLK, MRCLK* signals, and then read out of the FIFO buffers under control of the read pointer circuit 308 and multiplexers 310, 312 in groups at a slower rate determined by the CCLK signal.

The data words DW latched in the input capture registers 300, 302 are also provided to output capture registers 318, 320, respectively, and latched in the output capture registers responsive to master transmission clock signals MTCLK, MTCLK*. The MTCLK, MTCLK* signals are in the same clock domain as the MRCLK, MRCLK* signals, and would typically be derived from these clock signals. For example, the MRCLK and MRCLK* signals would typically be delayed to generate the MTCLK and MTCLK* signals, respectively, with the delay allowing the input capture registers 300, 302 to successfully latch the data words DW before the output capture registers 318, 320 latch these data words from the input capture registers.

Figure 4 is a signal timing diagram illustrating the operation of the memory hub 140 of Figure 3 in more detail in capturing and bypassing data words DW applied to the memory hub. In the example of Figure 4, the frequencies of the MRCLK, MRCLK*, MTCLK, MTCLK* signals are four times the frequency of the CCLK signal defining the clock domain of the memory hub 140. In operation, at a time T0 the input capture register 300 latches a first data word DW1 responsive to a rising edge of the MRCLK signal. This data word DW1 is latched by the output capture register 318 responsive to the MTCLK signal at a time T1 later. In this example, the MTCLK signal is delayed by a time T1-T2 relative to the MRCLK signal to ensure the data word DW1

is successfully stored in the input register 300 prior to the data word being latched by the output capture register 318. At a time T2, the input capture register 302 latches a second data word DW2 responsive to a rising edge of the MRCLK* signal (*i.e.*, a falling edge of the MRCLK signal as shown in Figure 4), and this data word is thereafter latched into the output capture register 320 at a time T3 responsive to a rising edge of the MTCLK* signal, which occurs at the same time as a falling edge of the MTCLK signal.

The input capture registers 300, 302 and output capture registers 318, 320 continue operating in this manner, each data word DW applied to the memory hub 140 being captured by the input capture registers and then applied to the output capture registers to thereby bypass the memory hub and provide these downstream data words to the next memory hub downstream. This capturing and bypassing occurs in the clock domain of the downstream high-speed links 204 and thus minimizes the delay introduced by each memory hub 140 in bypassing the downstream memory requests.

The data words DW captured in the input capture registers 300, 302 are also latched by the FIFO buffers 304, 306 responsive to the MRCLK, MRCLK* signals. The FIFO buffers 304, 306 are shown as being clocked by the MRCLK, MRCLK* signals for the sake of simplicity, and would actually be clocked by a signal derived from the MRCLK, MRCLK* signals, such as the MTCLK, MTCLK* signals, to ensure the data words DW are successfully stored in the input capture registers prior to the FIFO buffers latching the data words, as will be appreciated by those skilled in the art. Thus, each FIFO buffer 304, 306 latches the consecutive data words DW initially latched by the corresponding input capture register 300, 302.

The input capture registers 300, 302 and output capture registers 318, 320 continue operating in this manner to latch and bypass consecutive data words DW applied on the downstream high-speed link 204, as illustrated in Figure 4 at times T4-T9. In the example of Figure 4, each memory request is formed by 8 data words DW1-DW8 and each data word is 32 bits wide. After the last data word DW8 forming the memory request currently being transferred is latched in the input capture register

302 at time T9, this data word is latched into the FIFO buffer 306 at a time T10 responsive to the MRCLK* signal.

At this point, the entire memory request formed by the data words DW1-DW8 has been latched into the FIFO buffers 304, 306, with the buffer 304 storing data words DW1, DW3, DW5, DW7 and the buffer 306 storing data words DW2, DW4, DW6, DW8. At a time T11, the read capture read pointer circuit 308 applies the SEL signals to collectively output the data words DW1-DW8 from the multiplexers 310, 312 as the corresponding memory request. The memory controller 314 (Figure 3) thereafter processes the memory request from the multiplexers 310, 312.

While the memory request formed by the data words DW1-DW8 is being output from the multiplexers 310, 312, a next memory request is being applied on the high-speed link 204. At a time T12, the first data word DW1 of this next memory request is latched into the input capture register 300 responsive to the MRCLK signal. The memory hub 140 continues operating in this manner, with data words DW corresponding to a current memory request being applied on the high-speed link 204 being stored in the FIFO buffers 304, 306 while the previous memory request is output from the FIFO buffers. The capture read pointer circuit 308 develops the SEL signals to sequentially output the memory requests stored in the FIFO buffers 304, 306.

As previously mentioned, the depth of the FIFO buffers 304, 306 must be sufficient to allow the previous memory request to be output while a current memory request is being stored in the buffers. In the example of Figures 3 and 4, each of the buffers 304, 306 includes 12 storage locations, one for each data word DW. Thus, the buffers 304, 306 have a depth of 3 since they collectively store 3 consecutive memory requests. In this way, a current memory request may be stored a data word DW at a time in the FIFO buffers 304, 306 while the immediately prior memory request is stored in the buffers and the next prior memory request is output via multiplexers 310, 312 to the memory controller 314. The depth of the buffers 304, 306 may be varied, as will be appreciated by those skilled in the art. The buffers 304, 306 could have a minimum depth of 2, which would allow the currently applied memory request to be stored in the buffers as the prior memory request is output from the buffers. Using a depth of 3 or

more for the buffers 304, 306, however, eases the timing constraints on components in the physical reception port 202 (Figure 3), as will be appreciated by those skilled in the art.

The memory hub 140 of Figure 3 captures downstream data words DW and bypasses these data words to the next memory hub downstream in clock domain of the downstream high-speed links 204. Because this capturing and bypassing occurs in the faster clock domain of the downstream high-speed links 204, the delay introduced by each memory hub 140 in capturing and bypassing the downstream memory requests is minimized. Moreover, this approach simplifies the logic necessary to implement the local hub circuitry 214 (Figure 3), lowering the cost of each memory hub 140. In contrast, if each memory hub 140 determines whether a given memory request is directed to that hub and only bypasses requests not directed to the hub, the logic necessary to implement the local hub circuitry 214 would be much more complicated and thus the cost of each memory hub 140 would be higher. Each memory hub 140 would also introduce a greater delay of a given memory request with this approach, which would increase the latency of the system memory 102 (Figure 1) and is a potential drawback to a daisy-chain architecture, as previously discussed.

One skilled in the art will understand suitable circuitry for forming the components of the memory hubs 140, and will understand that the components implemented would use digital and analog circuitry.

In the preceding description, certain details were set forth to provide a sufficient understanding of the present invention. One skilled in the art will appreciate, however, that the invention may be practiced without these particular details. Furthermore, one skilled in the art will appreciate that the example embodiments described above do not limit the scope of the present invention, and will also understand that various equivalent embodiments or combinations of the disclosed example embodiments are within the scope of the present invention. Illustrative examples set forth above are intended only to further illustrate certain details of the various embodiments, and should not be interpreted as limiting the scope of the present invention. Also, in the description above the operation of well known components has

not been shown or described in detail to avoid unnecessarily obscuring the present invention. Finally, the invention is to be limited only by the appended claims, and is not limited to the described examples or embodiments of the invention.